

13

Express Number TB37788954545

08/262769  
PATENT

NSC1-54400

-1-

METHOD AND APPARATUS WHICH ALLOWS DEVICES  
WITH MULTIPLE PROTOCOL CAPABILITIES  
TO CONFIGURE TO A COMMON PROTOCOL CONFIGURATION

Brian C. Edem and Srinivas Kola

## 5 RELATED APPLICATIONS

*in 4/26/96*  
 This application is a Continuation-In-Part of <sup>copending</sup> application Serial No. 08/146,729, filed November 1, 1993, for "Network Link Detection and Generation" which is a Continuation-In-Part of application Serial No. 07/971,018, filed November 2, 1992, <sup>abandoned</sup> for "Network Link Endpoint Capability Detection."

*in 4/26/96*

## TECHNICAL FIELD OF THE INVENTION

The present invention relates to local area networks, and in particular, to methods and apparatus  
 15 which allows devices with multiple protocol capabilities to configure for a common protocol configuration.

## BACKGROUND OF THE INVENTION

The IEEE 802.3 standard defines Carrier Sense  
 20 Multiple Access/Collision Detection (CSMA/CD) as its Media Access Control (MAC) protocol for Ethernet local area networks (LAN's). A station connected to an Ethernet LAN may be fixedly configured to communicate with other stations of the LAN in a  
 25 single mode (e.g. 10BASE-T, 10BASE-T Full Duplex, 100BASE-TX, 100BASE-TX Full Duplex, and 100BASE-T4), or the communication mode of the station may be changeable to match the configuration of the other stations. In any case, the communication mode of a

2

particular station must match the communication mode of another station before communication can take place between the two stations.

5 The proliferation of CSMA/CD-based LANs will spawn a new generation of multi-function hubs and nodes. The trend among hub vendors towards "virtual networks" wherein ports can be switched between LANs will necessitate that, without an auto-negotiation solution such as is provided by the present  
10 invention, users and information services managers will face difficult configuration management problems in the future.

The necessity for providing in a station means for automatically detecting the communication  
15 capabilities of other stations in a network has been recognized. A method and apparatus for performing such automatic detection is the subject of copending patent application Serial No. 08/146,729, filed and assigned to National Semiconductor Corporation, the  
20 assignee of the present application. As described in the copending application, each of two connected stations communicates to the other station, via Fast Link Pulses, its respective communication capabilities. When each station has detected the  
25 other stations capabilities, the stations each automatically change to a configuration which provides the highest common denominator of capabilities.

30 However, the technology of LAN's is evolving such that manufacturers of LAN's will be providing stations which will be reconfigurable to support any of multiple protocols.

SUMMARY OF THE INVENTION

3

5 The present invention provides a method and apparatus which allows multi-protocol stations to automatically detect the protocols under which other stations can communicate, and for the connected stations to together select a common communication protocol.

10 The present invention solves the problem of configuring the growing available matrix for RJ-45 CSMA/CD compatible LANs by providing a simple, robust scheme to advertise, detect and resolve the capabilities of two stations connected with UTP. With the present invention, the upgrade occurs automatically and seamlessly from 10BASE-T to desired higher performance modes without requiring the  
15 presence of a management agent.

The present invention interoperates with all 10BASE-T compatible devices such that the installed network base will not be affected by the new generation of RJ-45 LAN technologies if NWay is  
20 adopted as a standard. NWay can be applied universally across node and hub products because the architecture is symmetric in nature and does not establish a master/slave hierarchy.

25 A low gate count and simple architecture make the present invention both easy to understand and cost effective. The cost and complexity are realistically bounded by providing a rich enough code space to meet the current projected higher performance mode and figure LAN technology needs.  
30 Although this proposal addresses the problem for CSMA/CD protocol LANs, it should be understood that other technologies can benefit from the use of this communication mechanism.

35 The present invention also provides a powerful management tool which allows:

4

1. Manual override of auto-negotiation;
2. Renegotiation on management intervention;
3. Remote Fault Sensing;
4. Parallel Advertisement of capabilities; and
5. Storage of remote capabilities.

The negotiation time is bounded for good links sufficiently to ensure that popular network protocol connections do not time out. Even in a noise UTP environment, the handshaking and redundant pattern comparison ensures robust operation.

A better understanding of the features and advantages of the invention will be obtained by reference to the following detailed description and accompanying drawings which set forth an illustrative embodiment in which the principles of the invention are utilized.

#### BRIEF DESCRIPTION OF THE FIGURES

~~Insert~~

#### DETAILED DESCRIPTION OF THE INVENTION

In the CSMA/CD protocol, as defined in IEEE 802.3, a 10BASE-T Normal Link Pulse (NLP) is applied on a network link segment to assist stations connected to the segment in determining the segment's integrity. Such a Link Pulse scheme is known in the art. As shown in Fig. 1, an NLP has a nominal pulse width of 100 ns. The pulses are transmitted every 16 ms +/- 8 ms while the segment is active, but not transmitting regular packet data (i.e. in an IDLE mode).

NLP's are used to control a station's entry to and exit from a Link\_Loss\_State. That is, a station enters Link\_Loss\_State when no NLP's are received by the station for more than link\_loss\_timer time. Link\_loss\_timer time typically ranges from 50 ms to

150 ms. A station exits Link\_Loss\_State when it detects a number of NLP's, typically two to ten.

The present invention substitutes a Fast Link Pulse Burst (FLP) for a single 10BASE-T NLP. An FLP burst, as shown in Fig. 2, consists of a maximum of 33 pulses. The 33 pulses are a combination of 17 clock pulses, with an optional data pulse being interleaved between each pair of adjacent clock pulses. In this manner, 16 data bits can be transmitted in each FLP burst -- a 1 data bit is encoded as a data pulse between a pair of adjacent clock pulses, and a 0 data bit is encoded as the absence of a data bit between the pair of adjacent clock pulses.

Individual pulses in an FLP burst have the same 100 ns width as NLP's, with both clock and data pulses having the same width. Clock pulses are spaced apart by 125  $\mu$ secs. Data bits, if present, are centered between a pair of adjacent clock pulses. Figs. 3, 4, and 5 summarize the FLP pulse and burst timing.

In accordance with an embodiment of the present invention, a method and apparatus are provided which allows a station at one end of a LAN link segment to advertise the protocols, and modes within those protocols, in which it can operate, and to likewise detect the protocols and modes in which a remote station at the other end of the link segment can operate (or at least those protocols and modes that the station at the other end of the link segment is advertising). The communication of modes takes place via the data communicated in a 16-bit "link code word" encoded in FLP bursts, as described above.

Fig. 6 illustrates a possible link code word encoding. Bits D0-D4 denote a "Selector Field" which

indicates a particular protocol under which the station advertising the link code word can communicate. Bit D5 denotes an "Acknowledge Bit". A station sets this bit in the link code word it is advertising when it has successfully received the remote station's link code word. Bits D6-D13 denote a "Technology Ability Field". These bits are mapped to individual technologies such that abilities are advertised in parallel for each selector field value. Also, multiple technologies may be advertised simultaneously by simultaneously setting the bits in the Technology Ability Field that correspond to the technologies. Fig. 7 shows a possible Technology Ability Field encoding when the selector field value indicates a CSMA/CD compatible protocol. Fig. 8 shows, for the selector field indicating a CSMA/CD compatible protocol, examples of how multiple technologies are advertised simultaneously. Bits D11-D14 are reserved for future use; these bits should be masked out upon reception of the link code word. Bit D15 is a "Next Page Bit" which provides a mechanism to transmit additional protocol and capability information in the link code word. That is, the Next Page Bit being set indicates that the other bits of the link code word contain protocol- and/or capability-specific information.

The present invention further provides a method and apparatus for using the protocol and capability information advertised in link code words communicated between two stations on a LAN link, of which one or both of the stations are multi-protocol/multi-capable, for each station to automatically detect the other stations advertised protocol/configuration capabilities and further automatically agree on a protocol/configuration under

7

which the stations can interoperate. Upon completion of the auto-agreement, each station configures its link, in accordance with the results of the auto-agreement, for interoperation with the other link.

5           Fig. 9 is a block-level diagram that illustrates how the protocol/configuration auto-agreement mechanism is implemented within each station. Preferably, the auto-agreement mechanism operates symmetrically so that a single implementation can be  
10           used in any multi-protocol/multi-capable station. In this way, both stations are on equal footing in the auto-negotiation process, with neither station being a "master" or "slave".

15           Fig. 9 illustrates the relationship between four state machines which are used to implement the auto-agreement mechanism within a station. An Arbitration State Machine controls the overall protocol/capability exchange process within the station, using and guiding the Transmit State Machine and the Receive State Machine. The Transmit State  
20           Machine controls the transmission of station protocol/capabilities using FLP bursts. The Receive State Machine controls the receipt of individual clock and data pulses within a Fast Link Pulse burst.  
25           A 10BASE-T Link Pulse State Machine controls the reception and transmission of NLP's when the station is communicating with a 10BASE-T station.

30           Finally, a Technology Dependent Link Integrity Test State Machine (or Machines) may co-exist with the auto-agreement scheme. For example, the Technology Dependent Link Integrity Test State Machine may be a 100BASE-TX Link Test-Fail State Machine which controls the reset logic for the Convergence Sub-Layer of the 100BASE-TX standard.

When in Link\_Fail, the Idle Line State is continuously sent.

Each of the auto-agreement state machines are now described in detail.

5           The arbitration state machine is shown in  
Fig. 10. The purpose of this state machine is to  
control the FLP transmit state machine and determine  
if a remote station has its technology abilities,  
either through FLPs or through a technology-dependent  
10           signalling scheme. At power on, or upon the  
assertion of technology-dependent link-fail  
condition, this state machine moves into Capability  
Detect state to allow the FLP transmit state machine  
to start sending FLP bursts containing the local  
15           station's technology abilities to the far-end  
station. The results of FLP receive state machine  
are to be used to determine technology ability match.

A technology ability match occurs when this  
state machine is in Ability Detect state, and three  
20           consecutive FLP bursts contain the same pattern  
ignoring the acknowledge bit. Once a capability  
match occurs, the FLP arbitration state machine moves  
into Acknowledge Detect state.

In Acknowledge Detect state, it transmits the  
25           same data pattern with the acknowledge bit set, to  
indicate to the far-end station that it has captured  
its technology abilities.

In the case of unexplained loss of FLP bursts  
coming from the far-end station, under which the FLP  
30           receive state machine goes to IDLE state, this state  
machine goes back to Ability Detect state to restart  
negotiation from scratch. While remaining in the  
Acknowledge Detect state, it waits to receive  
acknowledge\_match from the far-end station.

35           Acknowledge\_match occurs when two out of three



consecutive FLP bursts received have the acknowledge bit set. In order to guarantee that the far-end station has completed acknowledge\_match, a minimum of 4-6 acknowledge patterns are transmitted.

5 Subsequently, it transitions to FLP\_link\_good state. After this point, the technology-specific link signalling scheme takes over, if needed for the specific technology (e.g., 100BASE-TX, 100BASE-T4). If this technology-specific link signalling scheme  
10 does not successfully complete in a technology determined period of time, then technology\_link\_fail is asserted, and the Arbitration State Machine starts renegotiation.

15 If the far-end station does not detect and transmit FLP bursts, and instead sends a technology-dependent signal (e.g., 10BASE-T link pulses, or 100BASE-TX Idle Line State transmissions), the technology-dependent\_link\_good signal thereby halting FLP burst transmits.

20 In addition, upon successful establishment of technology-dependent link, the technology-dependent link integrity test state machine goes to link fail and the arbitration state machine goes back to the Ability Detect state.

25 A higher-level management agent can start a renegotiation of technology abilities with the far-end station by asserting the renegotiate signal. The state machine moves into Transmit Disable state to wait for the far-end station to go into a link-fail  
30 state and then starts FLP burst transmissions.

The Transmit state machine is shown in Fig. 11. The purpose of this state machine is to transmit a sequence of 33 fast link-pulses to the far-end station, when instructed by the arbitration state  
35 machine. The idle stats is entered at power-on and

this state machine transitions to this state if the arbitration state machine is in the FLP\_link\_good state.

5 The transmit\_link\_pulse\_timer (14 +/-8 msec) is used to separate two consecutive fast link-pulse bursts transmitted.

10 If the arbitration state machine is in the process of sending the capability to the far-end station, then the transmit state machine enters into the Transmit\_Capability state. If acknowledgement is to be sent, then the state machine enters Transmit\_Acknowledge state to initialize ack\_cnt (4-16), and then it moves to Transmit\_Capability state. In this state, the bit\_cnt is started to keep track  
15 of the 16 bits of data to be transmitted.

The state machine then alternates between the Transmit\_Clock\_Bit state and Transmit\_Data\_Bit state to transmit clock bit and to transmit data bit (if the data bit to be transmitted is logic one)  
20 respectively. The interval\_timer (62.5 +/-15  $\mu$ s) is used to separate a clock bet and a data bit.

Once all 16 bits of data are transmitted, if the state machine is in the midst of sending acknowledgement patterns then the state machine moves  
25 to the Transmit\_Count\_Ack state. However, after 16 bits of data transmission, if the state machine is not in the midst of sending acknowledgement patterns, then the state machine moves to idle directly.

30 The state machine returns to idle after a 16-bit capability has been transmitted. In the Transmit\_Count\_Ack state, if 4-6 acknowledgement patterns have been sent, it goes back to idle, otherwise it waits for 14+/-8 msec and moves to Transmit\_Capability state to send the next burst of  
35 16-bit acknowledgement pattern.

The Receive state machine is shown in Fig. 12. This state machine is used to identify that the link pulses received are fast-link-pulse bursts, and if so, to store the embedded 16-bit data in a shift-  
5 register. A burst of thirty-three link pulses separated by 62.5 uS conveys the capability of the far-end station. A link pulse is considered part of a FLP burst if it occurs within a certain time of the previous link-pulse. It is detected by the use of  
10 the two timers: FLP\_test\_min\_timer and FLP\_test\_max\_timer. The purpose of the FLP\_test\_min\_timer is to mask out any noise or ringing effects on the line. The FLP\_test\_max\_timer is used to determine whether the next link-pulse  
15 received is within a window to be of fast-link-pulse category.

Fig. 13 illustrates the application of the FLP Detect Timers (FLP\_test\_min/max\_timers). The Link Receive state machine goes to the idle state at  
20 power-on or when the arbitration state machine forces it to this state. When the Link Receive state machine is in idle state, a reception of a link-pulse causes a transition to the Link\_Pulse\_Detect state.

At this point, both the FLP\_test\_min\_timer and  
25 FLP\_test\_max\_timer are started. The FLP\_test\_min\_timer has a value of 5 to 25  $\mu$ s. The FLP\_test\_max\_timer has a value of 165 to 185  $\mu$ s. If another link-pulse is received in the window of time such that the FLP\_test\_min\_timer has timed out, but  
30 the FLP\_test\_max\_timer has not completed, the state machine makes a transition to Link\_Pulse\_Count state.

In the Link\_Pulse\_Count state, FLP\_count is incremented. If consecutive number of fast link pulses are received, i.e., the FLP\_count equals  
35 FLP\_count\_max (6 to 8), then the far-end station is

considered FLP capable, and all Technology-Specific link integrity test state machines (e.g., 10BASE-T link\_integrity\_test\_state machine) are forced into Freeze state.

5           The Receive state machine then goes into a FLP\_Pass state to ignore the rest of the incoming fast link-pulse burst. This is accomplished by waiting for the idle interval after the last link pulse of the current burst, i.e., an idle period of  
10       165 to 185  $\mu$ s has been observed. At this time, the state machine transitions into FLP-Capture state, and is ready to capture a complete burst of fast link pulses from the far-end station.

15           In the FLP-capture state, the link\_test\_max\_timer (25-150 msec.) is started to insure that link pulse bursts are received within this window. When this timer expires, the state machine moves to the idle state to indicate the absence of FLP bursts.

20           Fig. 14 illustrates the operation of the FLP Burst Timer (Link\_test\_min/max\_timers). The first link-pulse received in the FLP\_capture state is interpreted as a clock bit. Two consecutive clock bits can envelop a data bit. The presence of a data  
25       bit is interpreted as a logic one for the corresponding data bit field of the shift register. the absence of the data bit is interpreted as a logic zero for the corresponding data bit field of the shift register. Two consecutive clock pulses are  
30       separated by 125 +/- 15  $\mu$ s. The separation of a subsequent data bit from a clock bit is 62.5 +/- 15  $\mu$ s. These margins are designed to be symmetric for both local and far-end stations.

35           In the FLP\_Clock state, the data\_detect\_min\_timer (15 to 47  $\mu$ s) and

data\_detect\_max\_timer (78 to 110  $\mu$ s) are started. If  
a link-pulse is received when the  
data\_detect\_max\_timer has completed and  
data\_detect\_max\_timer is not, it is interpreted as a  
5 data-bit of logic one. On the other hand, if the  
next clock\_bit arrives, as indicated by  
data\_detect\_max\_timer\_done, then the embedded data  
bit is assumed to be zero. In the FLP\_Clock state,  
the link\_test\_min\_timer (5 to 7 msec) is also  
10 started; this timer is used to separate two  
consecutive fast-link pulse bursts.

Fig. 15 illustrates the operation of FLP Data  
Detect Timers (Data\_detect\_min/max\_timers). The  
FLP\_Data\_0 state, a logic zero is shifted into the  
15 shift register. In FLP\_Data\_1 state, a logic one is  
shifted into the shift register and upon the  
reception of the next clock-bit, the state machine  
moves back to FLP\_Clock state. Either upon the  
reception of the last clock bit, or due to a bit  
20 error corrupting the last clock but, the state  
machine moves to FLP\_Capture state.

Fig. 16 is a table which summarizes the timer  
minimum and maximum values.

The only technology-specific state machine that  
25 is required is the 10BASE-T link integrity test state  
machine. This state machine is required for  
compatibility with existing 10BASE-T nodes.

Two modifications must be made to the original  
10BASE-T link integrity test state machine. The  
30 first modification is that the state machine must  
power-up in the Link\_Test\_Fail\_Reset state.

The second modification needed to the original  
10BASE-T link integrity test state machine is to add  
another state called Freeze 10BASE-T. This state is  
35 entered upon the recognition of the far-end station's

FLP capability and remains in this state during the negotiation phase. After the NWay negotiation is successfully completed, and the highest common denominator is determined to be 10BASE-T, the  
5 10BASE-T Link integrity test state machine moves into the Link\_test\_pass\_state. The state machine remains in the Freeze\_10BASE-T state if the mode of communication selected is not 10BASE-T.

10 Additional technology-specific state machines can be added to implement technology-specific link testing.

The Peripheral Logic Block/Management Interface contains the following:

- 15 1) Three 16-bit shift registers to hold the received 16-bit capability information from the far-end station
- 2) a 2-bit counter to select the current shift-register of the above-mentioned three, in which the data-bits are being shifted,
- 20 3) a 16-bit local capability register,
- 4) a 16-bit far-end capability register,
- 5) combinational gates to generate:
  - capability\_match signal for arbitration state machine,
  - 25 acknowledge\_match signal for arbitration state machine,
  - highest-common-denominator.

30 The next two sections provide the Technology Bit Field Assignments and Priority Resolution Table for the Message Selector.

Two nodes can have multiple abilities in common. In accordance with one embodiment of the present invention a prioritization scheme exists to ensure that the highest common denominator ability is

chosen. For example, the following priority may be used (from highest to lowest priority):

- 1) 100BASE-TX Full Duplex;
- 2) 100BASE-T4;
- 5 3) 100BASE-TX;
- 4) 10BASE-T Full Duplex; and
- 5) 10BASE-T.

The rationale for this hierarchy is straightforward. 10BASE-T is the lowest common denominator and therefore that the lowest priority. Full Duplex solutions are always higher in priority than their Half Duplex counterparts. 100BASE-T4 is ahead of 100BASE-TX because 100BASE-T4 runs across a broader spectrum of copper cabling.

#### 15 Operational Examples

The following operational examples highlight the steps that are taken when attempting to establish a network connection between two nodes. Only three general cases are necessary to demonstrate all the different combinations of operation that will ever occur. The three cases are any auto-detect capable device to 10BASE-T, any auto-detect capable device to another auto-detect capable device, and any auto-detect capable device to a non-auto-detect capable device. All these examples fall into one of these three cases.

Case 1: NWay Capable <---> 10BASE-T in Link\_Fail State (Fig. 17).

This example highlights the invention's 100% backward compatibility with 10BASE-T. The 10BASE-T station in this example begins in the Link\_Fail state and is transmitting Normal Link Pulses. The auto-detect capable station supports an arbitrary

100Mb/s/10BASE-T capable or 100Mb/s only capable node. Initially the NWay capable station comes up in the Link\_Fail state and transmits FLP bursts to advertise its technology ability.

5           The normal 10BASE-T node receives the FLP's and remains in Link\_Fail state because the timing of the FLP's do not allow a normal 10BASE-T station to misinterpret FLPs as Normal Link Pulses. To understand this, there are two main cases to consider.

10           In the first case, shown in Fig. 18, the 10BASE-T only station begins receiving FLPs in between bursts. The link\_test\_min\_timer (2-7ms) expires and the first pulse in an FLP burst gets counted as a good link pulse. However, subsequent

15           pulses in the FLP burst arrive before the link\_test\_min\_timer expires for a second time which resets the good link pulse counter to zero.

20           In the second case, shown in Fig. 19, the 10BASE-T only station begins receiving FLPs just before the last pulse in an FLP burst. This pulse is not counted as a good link pulse because the link\_test\_min\_timer has not expired yet. This pulse therefore keeps the good link pulse counter reset at

25           zero. This case then proceeds like the first case.

30           No scenario causes the good link pulse counter to count higher than 1, which is less than the 2 to 10 required good link pulse specification, so the 10BASE-T only station will never mistakenly enter the Link\_Pass state.

35           The 10BASE-T Link Integrity Test state machine is a required part of the auto-detect, regardless of 10BASE-T data transmit/receive capability. The NWay capable 100/10 node recognizes the NLPs being sent by the normal 10BASE-T node, switches over to 10BASE-T



operation, and sends NLPs. The two stations then both communicate in 10BASE-T mode. If the auto-detect capable station is 100Mb/s only, then it must still recognize 10BASE-T NLPs being sent. In  
5 response to NLPs, FLP burst transmission is halted and no NLPs are transmitted by the 100Mb/s only station.

Case 2: Auto-Detect Capable <---> 10BASE-T only in Link\_Fail State (Fig. 20).

10 This example highlights auto-detect's 100% backward compatibility with 10BASE-T. The 10BASE-T station in this example begins in Link\_Pass state and is transmitting Normal Link Pulses or normal 10BASE-T traffic. The auto-detect capable station is an  
15 arbitrary 100Mb/s/10BASE-T capable or 100Mb/s capable station. Initially the auto-detect capable station comes up in Link\_Fail state and transmits Fast Link Pulses to advertise its technology abilities.

20 The Normal 10BASE-T node receives the Fast Link Pulses and always remains in Link\_Pass state because the link\_loss\_timer does not expire.

The auto-detect capable node's 10BASE-T specific link integrity test state machine identifies the link  
25 as being good due to receipt of 10BASE-T traffic or normal link pulses.

If no 10BASE-T data transmit/receive capability exists, then FLP bursts are halted, and no NLPs are transmitted by the auto-detect capable station,  
30 otherwise NLPs are transmitted and the link is established.

If some 10BASE-T Link Integrity Test state machine implementations transition from Link Pass state to Link Fail state due to FLPs, we are back to

the previous auto-detect capable <---> 10BASE-T in Link Fail state example.

Case 3: Auto-Detect Capable Connection With Multiple Common Capabilities (Fig. 21)

5           This example highlights auto-detect capable station to auto-detect capable station auto-negotiation when multiple common technology abilities are supported by both stations. In this case it is assumed that both stations support Full Duplex 100BASE-TX, 100BASE-T4 and 10BASE-T.

10           Both nodes come up in Link\_Fail state and transmit FLP bursts encoded with their technology abilities. Each node learns its partners' technology abilities. In this case each node learns that it has Full Duplex 100BASE-TX, 100BASE-T4 and 10BASE-T in common.

15           The built-in priority resolution table in each node indicates that full Duplex 100BASE-TX is the highest common denominator technology ability, so both nodes switch over to this technology ability and begin transmission.

Case 4: Auto-Detect Capable Connection Between 100BASE-T4 Nodes (Fig. 22)

25           This example highlights a connection between two 100BASE-T4 auto-detect capable stations. In this case it is assumed that each station supports 100BASE-T4 as its highest common denominator technology ability.

30           At this time the 100BASE-T4 technology ability can proceed through its own ability-specific link integrity test. This test can then do a specific test to insure that the two pairs of cable not used for auto-detect signalling are intact.

In one aspect of the present invention, redundant transmissions are relied upon to ensure that the transmitted 16-bit pattern is correctly received by the remote station. That is, the remote station must receive an identical 16-bit pattern three times in a row before the acknowledgement pattern is transmitted. The acknowledgement pattern is the same 16-bit word with the acknowledge bit set. The reception of 2 out of 3 acknowledgement patterns completes the handshake. Thus, it is assured that the correct configuration information is exchanged.

The present invention provides an efficient mechanism for point-to-point communication that is efficient in terms of the amount of logic gates required. Regardless of the number of technology abilities supported, a total of 3 state machines, 6 timers, 3 counters, and some peripheral logic is needed. An analysis of the architecture shows that the design will require approximately 400 cells to implement. In addition, the present invention has been architected as a simple mechanism which allows easy comprehension by silicon suppliers.

It should be noted that this level of complexity remains fixed as new technologies emerge and need to coexist with the growing installed based of RJ-45 LANs. Alternative schemes that require separate and complex state machines for each pattern will be more complex and expensive to implement, especially as new LAN options are added.

The present invention can maintain compatibility when new options are added. As new CSMA/CD-compatible LAN technologies enter the market a reserved bit may be assigned to each technology. The new technology will be inserted into an updated priority table, as detailed in the Data Bit Field

description. The relative hierarchy of the existing technologies need not change, thus providing backward compatibility. In preferred embodiments, the reserved bits are forced to zeros. This guarantees that devices implemented using the current priority table will be forwarded compatible with future devices using an updated priority table.

The present invention allows a station to advertise its complete set of CSMA/CD capabilities in one burst, a feature referred to as its "parallel advertising" capability. Advertising the complete set of capabilities in one pass allows a station to immediately discover all capabilities in one pass and thus immediately discover all capabilities of the remote station. These capabilities can then be saved for passage to a management agent.

In accordance with a further embodiment, although a pair of stations default to their highest common denominator capability, other modes of interoperation may exist. For a variety of reasons, the network management agent may wish to reconfigure the link.

#### Protocol Arbitration and Resolution Algorithm

In accordance with a further embodiment of the present invention, a Protocol Arbitration and Resolution Algorithm (PARA) algorithm allows for arbitration and resolution for capabilities within a given protocol and, further, between the protocols themselves until a common protocol and ability are identified. If no common ability is identified, then the link will not be established.

Solution 1:

The PARA's shown in flow-chart form in Fig. 23.

**Start of Algorithm (label A):**

On the reception of 3 consecutive and identical link code words (label A) all the four timers in the PARA are reset (label A).

**5      Protocol Identifiers Do Not Match (label B)**

10      If the Protocol Identifiers (PID's) do not match (label B) then it will be verified if the local node can support other protocols (label 2). If another PID can be supported (label 2) then it will be set (label 3) in the 16 bit link code word or else no change will be made to the PID. Control returns back to the beginning of the algorithm (label A). If the PID of the local node is not the lowest in the negotiating pair, then the local node will wait until 15      the timer 1 (Slave\_code\_word\_Change\_timer) is done (label 4) for the other node to change its PID and advertise. If no change is observed in the link code word, then it knows that the other node is incapable of protocol arbitration and becomes the arbitration 20      master (i.e, jumps to label 2). If there is a change in the link code word the local node knows that the link partner changed the PID and will have to renegotiate (label 5) and hence will return control back to beginning of the algorithm (label A). (Note: 25      this accommodates for devices with multiple protocols with no protocol arbitration function and also for devices with multiple protocols but with the arbitration function disabled manually or by software).

**30      Protocol Identifiers Match (label B):**

If the protocol identifier's (PID) match (label B) then the capability fields of local node and its

22

link partner are compared. If there are common capabilities then the highest common denominator amongst the common capabilities is chosen as specified in NYway (label 6).

5           If the PID's match and there are no common abilities then both the link partners will evaluate to see who has the lowest ability field (label C).

**Lower Ability With PID Match (label C):**

10           If the local node has the lowest ability, then it will be verified if it has other abilities (label 7) and if found true the new capability or abilities can be set in the 16 bit link code word and return to beginning of algorithm. On the other hand if no other abilities are present then the local node will  
15           wait until timer 2 (Master\_ability\_change\_timer) is done (label 8) giving time for the remote end (label 11) to advertise ability bits which it might not have previously advertised. If there is no change in the link code word, then the given PID is exhausted and  
20           hence a new PID will be set and negotiation will start at the beginning of the algorithm (label A).

**Higher Ability With PID Match (label C):**

25           In this case the remote link partner will be the negotiation master. Local node (label 11) will wait until timer 1 (Slave\_code\_word\_change\_timer) is done, to let the master change the ability bits and advertise the link code word. If there is a change in the bit field before this timer is done, then the master is still in control and negotiating. If no  
30           change is seen when the timer is done then the local node takes initiative/control from the remote master temporarily (label 12). The local node sets additional ability bits in the link code word if it

has any previously unadvertised abilities it might want to advertise now. Control will be transferred to the beginning of the algorithm. On the other hand if it does not have any new ability bits to advertise then it will wait until the timer 3

(Slave\_Negotiation\_timer) is done (label 13). Before the timer 3 is done the remote master waits until its timer 2 (Master\_ability\_change\_timer) is done, knows that all the feasible ability matching options are exhausted in both the nodes and hence tries to change its PID as described in the section above (label 9) before losing the Negotiation master privilege. If the remote master has no PID arbitration capability, it will yield mastership to the local node. When the timer 3 (Slave\_Negotiation\_timer) is done at the local end the local node takes up negotiation control and advertises a new PID if it is capable.

---

Timer 1: Slave\_code\_word\_change\_timer 11x16=176ms

Timer 2: Master\_code\_word\_change\_timer 22x16=352ms  
Master\_ability\_change\_timer

Timer 3: Slave\_Negotiation\_timer 33x16=528ms

Timer 4: Master\_Negotiation\_timer 36x16=576ms.

---

The following is an example for the application of "Protocol Arbitration and Resolution Algorithm" described above.

The following example shows how two systems can find a common protocol and a common set of capabilities. Assume a link with two endpoints,

called X and Y. These Endpoints have the following protocols and abilities:

Endpoint X:

Protocol 1:

5 Abilities A B C

Protocol 3:

Abilities G H I

Endpoint Y:

Protocol 2:

10 Abilities D E F

Protocol 3:

Abilities I J K

15 Each protocol can represent a different encoding scheme, transmission rate, or access mechanism. Each ability can represent different configurations, modes of operation, or optional abilities.

20 Assume that X initially advertises Protocol 1, abilities A, B, C, (represented by the notation 1-ABC), and Y advertises 2-DEF. Since these protocols are different, they can not interoperate (label B). Since X is advertising a protocol with the smallest encoded value, it is X's responsibility to select a new protocol (X:label 2; Y:label 4). Since X can not support Protocol 2, it will advertise  
25 Protocol 3. When X advertises Protocol 3, it decides that it would prefer to use Abilities G or H, and will initially only advertise 3-GH (X:label 4; Y:label 4).

30 Endpoint Y, now being the endpoint with the lowest encoded value for protocol, has two options --



if it were capable of communicating using protocol 1,  
it could advertise it, forcing X to return to  
protocol 1, or, as in this case, it will advertise  
protocol 3-IJK (Y is willing to use any of its  
capabilities).

The protocols now match, but the intersection of  
the two sets of abilities is empty. If we assume  
that when the abilities (advertised by Y) are encoded  
that 3-IJK will represent a lower value than 3-GH  
(advertised by X), it will be endpoint Y's  
responsibility to change capabilities. Endpoint Y  
has no other options, and makes no changes (X:label  
11; Y:label 8).

Endpoint X reaches a timeout 1 (i.e., when timer  
1 or slave\_code\_word\_change\_timer is done), realizing  
that Y is unable to proceed, so X will change its  
advertisement (X:label 12) to include the additional  
ability of I (control transfer afterwards to X:label  
A; Y:label A).

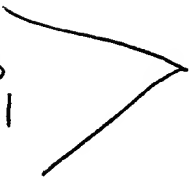
X's advertisement is now 3-GHI, which contains  
the common element 1 with Y's advertisement of 3-  
IJK, allowing the completion of arbitration (X:label  
6; Y:label 6).

If endpoint X had no other capabilities (i.e., X  
has only abilities 3-GH), it too would make no  
changes to its capabilities, allowing timeout2 (i.e.,  
timer 2 or Master\_ability\_change\_timer is done) to be  
reached by endpoint Y. Upon reaching this timeout2,  
Y has the opportunity to change to another protocol.  
In this case, there are no other protocols to be

tried, and Y will make no changes to its advertisement.

5 A third timeout (timer 3 or Slave\_negotiation\_timer is done) is reached by X, indicating that Y was unable to select a new protocol, giving X the opportunity to select a new protocol. This mechanism allows X to initially ignore one of Y's protocol until X determined that there were no other common protocols.

INS  
BI



**Solution H:**

This is a second method to achieve the same results but the arbitration is not done until all the Protocol capabilities and technology abilities are exchanged between the two link partners.

The bits A7-A0 in the link code word in Fig. 6 has different definitions for different PID's or selector values in the link code word (D4-D0). A bit definition for a new PID/selector value of 00000 will be defined and can be used to initiate negotiation across devices with multiple protocols. The bit definition in this case is defined in Fig. 24. *The next page bit can be used to transfer vendor ID.*

The link partners X and Y using this scheme for negotiate is described below:

Node X sends the link code word with selector/PID set to 00000, ack=0, series end bit = 0, and a four bit binary encoded value of the number of protocols supported (to node Y).

If Node Y is not multiprotocol capable then it will keep sending the link code word with the only PID it supports or can handle. On sensing this X will identify that Y is not multiprotocol capable and will advertise with the same PID as of node Y, if X can handle that particular protocol. If there are common technology abilities between X and Y then a link will be established. If X is not capable of the protocol advertised by Y then link cannot be established.

If node Y is multiprotocol capable then it also responds with a PID=00000, the number of protocols it can support, and other information about itself through the link code word. As specified in the earlier portion of the document the ack bit is set on the reception of 4-6 consecutive and consistent link code words. Both nodes begin to transmit their capabilities starting with the lowest PID value they support until all the protocol capabilities along with the technology ability bits are transmitted. It is possible that one of the nodes with fewer protocol abilities can finish its transmission ahead of its link partner. In such a case it will begin transmitting PID=00000 with the series end bit set to a "1". The node that is lagging will soon catch up and will also begin transmitting PID=00000 and series end bit = 1. Since both nodes have advertised the end of transmission, this signifies the end of capability exchange between the two nodes. (If all the protocol information received matches the binary encoded number of protocols identified in the first code word transmission then there is no loss of information, else retransmission will be done from the beginning.) From the beginning to the end of this negotiation with PID=00000, NWay will not configure to the highest common denominator in any protocol (it will be in a partial freeze state).

Now that all the protocol capabilities are known the media access units (MAU's) in both the nodes will look up a common prioritization table to identify the highest common denominator common to both the nodes..

Prioritization table gives the order in which a node prefers to configure based on the protocols and the abilities within that protocol. Example:

- 1) protocol A, ability 1
- 2) protocol B, ability 5
- 3) protocol A, ability 2
- 4) protocol B, ability 2

It is obvious that the abilities within a protocol can be mixed in the prioritization table with the abilities from other protocols. This gives tremendous flexibility beyond simply prioritizing the protocols. This also allows network management to mask out certain abilities such that the chosen ability is the most optimal for the kind of service (like data transmission, reliable and isochronous for video/multimedia or lowest cost service etc.) required.

Thus, it can be seen that the protocol resolution and arbitration algorithm allows nodes supporting multiple protocols (e.g as shown in Fig. 25) to coexist on a single network. It allows a node (e.g. a computer) to scan the protocol capabilities of other nodes connected to it on the network and configure to a common and desirable protocol configuration.

*cu*  
*4/26/96*  
10 ~~Attached to this specification are the following~~  
two appendices, both of which are to be considered as  
an intergral part of this specification:

15 Appendix 1) Draft Supplement to IEEE Std 802.3  
P802.3ug/D1 100 BASE-T 28. Physical Layer Link  
Signalling for 10 & 100 Mb/s Auto-Negotiation on  
Twisted Pair

Appendix 2) IEEE Link Task Force Autodetect,  
~~Specification for NWAY Autodetect.~~

20 It should be understood that various  
alternatives to the embodiments of the invention  
described herein may be employed in practicing the  
invention. In particular, the invention has  
application beyond local area networks; it has  
application wherever there is communication between  
two stations.

25 It is intended that the following claims define  
the scope of the invention and that methods and  
apparatus within the scope of these claims and their  
equivalents be covered thereby.